MeMOTR: Long-Term Memory-Augmented Transformer for Multi-Object Tracking

Ruopeng Gao[1] Limin Wang[1,2,✉]
[1]State Key Laboratory for Novel Software Technology, Nanjing University [2]Shanghai AI Lab

Abstract

As a video task, Multiple Object Tracking (MOT) is expected to capture temporal information of targets effectively. Unfortunately, most existing methods only explicitly exploit the object features between adjacent frames, while lacking the capacity to model long-term temporal information. In this paper, we propose MeMOTR, a long-term memory-augmented Transformer for multi-object tracking. Our method is able to make the same object's track embedding more stable and distinguishable by leveraging long-term memory injection with a customized memory-attention layer. This significantly improves the target association ability of our model. Experimental results on DanceTrack show that MeMOTR impressively surpasses the state-of-the-art method by 7.9% and 13.0% on HOTA and AssA metrics, respectively. Furthermore, our model also outperforms other Transformer-based methods on association performance on MOT17 and generalizes well on BDD100K. Code is available at https://github.com/MCG-NJU/MeMOTR.

1. Introduction

Multi-Object Tracking (MOT) [8, 23, 29] aims to detect multiple objects and maintain their identities in a video stream. MOT can be applied to numerous downstream tasks, such as action recognition [7], behavior analysis [16], and so on. It is also an important technique for real-world applications, *e.g.*, autonomous driving and surveillance.

According to the definition of MOT, this task can be formally divided into two parts: object detection and association. For a long time, pedestrian tracking datasets (like MOT17 [23]) have had mainstream domination in the community. However, these datasets have insufficient challenges in target association because of their almost linear motion pattern. Therefore, tracking-by-detection methods [5, 32, 43] achieve the state-of-the-art performance of MOT for several years. They first adopt a robust object detector (*e.g.*, YOLOX [12]) to independently localize the objects in each frame and associate them with IoU [3, 40] or ReID features [26]. However, associating targets becomes a critical challenge in some complex scenarios, like group dancers [29] and sports players [8, 13]. These similar appearances and erratic movements may cause existing methods to fail. Recently, Transformer-based tracking methods [22, 42] have introduced a new fully-end-to-end MOT paradigm. Through the interaction and progressive decoding of detect and track queries in Transformer, they simultaneously complete detection and tracking. This paradigm is expected to have greater potential for object association due to the flexibility of Transformer, especially in the above complex scenes.

Although these Transformer-based methods achieve excellent performance, they still struggle with some complicated issues, such as analogous appearances, irregular motion patterns, and long-term occlusions. We hypothesize that more intelligent leverage of temporal information can provide the tracker a more effective and robust representation for each tracked target, thereby relieving the above issues and boosting the tracking performance. Unfortunately, most previous methods [22, 42] only exploit the image or object features between two adjacent frames, which lacking the utilization of long-term temporal information.

Based on the analysis above, in this paper, we focus on leveraging temporal information by proposing a long-term **Me**mory-augmented **M**ulti-**O**bject **T**racking method with T**R**ansformer, coined as **MeMOTR**. We exploit detect and track embeddings to localize newborn and tracked objects via a Transformer Decoder, respectively. Our model maintains a long-term memory with the exponential recursion update algorithm [28] for each tracked object. Afterward, we inject this memory into the track embedding, reducing its abrupt changes and thus improving the model association ability. As multiple tracked targets exist in a video stream, we apply a memory-attention layer to produce a more distinguishable representation. Besides, we present an adaptive aggregation to fuse the object feature from two adjacent frames to improve tracking robustness.

In addition, we argue that the learnable detection query

✉ : Corresponding author (lmwang@nju.edu.cn).

in DETR [6] has no semantic information about specific objects. However, the track query in Transformer-based MOT methods like MOTR [42] carries information about a tracked object. This difference will cause a semantic information gap and thus degrade the final tracking performance. Therefore, to overcome this issue, we use a light decoder to perform preliminary object detection, which outputs the detect embedding with specific semantics. Then we jointly input detect and track embeddings into the subsequent decoder to make MeMOTR tracking results more precise.

We mainly evaluate our method on the DanceTrack dataset [29] because of its serious association challenge. Experimental results show that our method achieves the state-of-the-art performance on this challenging DanceTrack dataset, especially on association metrics (*e.g.*, AssA, IDF1). We also evaluate our model on the traditional pedestrian tracking dataset of MOT17 [23] and the multi-categories tracking dataset of BDD100K [41]. In addition, we perform extensive ablation studies further demonstrate the effectiveness of our designs.

2. Related Work

Tracking-by-Detection is a widely used MOT paradigm that has recently dominated the community. These methods always get trajectories by associating a given set of detections in a streaming video.

The objects in classic pedestrian tracking scenarios [9, 23] always have different appearances and regular motion patterns. Therefore, appearance matching and linear motion estimation are widely used to match targets in consecutive frames. SORT [3] uses the Intersection-over-Union (IoU) to match predictions of the Kalman filter [33] and detected boxes. Deep-SORT [34] applies an additional network to extract target features, then utilizes cosine distances for matching besides motion consideration in SORT [3]. JDE [32], FairMOT [44], and Unicorn [38] further explore the architecture of appearance embedding and matching. ByteTrack [43] employs a robust detector based on YOLOX [12] and reuses low-confidence detections to enhance the association ability. Furthermore, OC-SORT [5] improves SORT [3] by rehabilitating lost targets. In recent years, as a trendy framework in vision tasks, some studies [35, 47] have also applied Transformers to match detection bounding boxes. Moreover, Dendorfer *et al*. [10] attempt to model pedestrian trajectories by leveraging more complex motion estimation methods (like S-GAN [14]) from the trajectory prediction task.

The methods described above have powerful detection capabilities due to their robust detectors. However, although such methods have achieved outstanding performance in pedestrian tracking datasets, they are mediocre at dealing with more complex scenarios having irregular movements. These unforeseeable motion patterns will cause the trajectory estimation and prediction module to fail.

Tracking-by-Query usually does not require additional post-processing to associate detection results. Unlike the tracking-by-detection paradigm mentioned above, tracking-by-query methods apply the track query to decode the location of tracked objects progressively.

Inspired by DETR-family [6], most of these methods [22, 42] leverage the learnable object query to perform newborn object detection, while the track query localizes the position of tracked objects. TransTrack [30] builds a siamese network for detection and tracking, then applies an IoU matching to produce newborn targets. TrackFormer [22] utilizes the same Transformer decoder for both detection and tracking, then employs a non-maximum suppression (NMS) with a high IoU threshold to remove strongly overlapping duplicate bounding boxes. MOTR [42] builds an elegant and fully end-to-end Transformer for multi-object tracking. This paradigm performs excellently in dealing with irregular movements due to the flexibility of query-based design. Furthermore, MQT [17] employs different queries to represent one tracked object and cares more about class-agnostic tracking.

However, current query-based methods typically exploit the information of adjacent frames (query [42] or feature [22] fusion). Although the track query can be continuously updated over time, most methods still do not explicitly exploit longer temporal information. Cai *et al*. [4] explore a large memory bank to benefit from time-related knowledge but suffer enormous storage costs. In order to use long-term information, we propose a long-term memory to stabilize the tracked object feature over time and a memory-attention layer for a more distinguishable representation. Our experiments further approve that this approach significantly improves association performance in MOT.

3. Method

3.1. Overview

We propose the **MeMOTR**, a long-term memory-augmented Transformer for multi-object tracking. Different from most existing methods [22, 42] that only explicitly utilize the states of tracked objects between adjacent frames, our core contribution is to build a *long-term memory* (in Section 3.3) that maintains the long-term temporal feature for each tracked target, together with a *temporal interaction module (TIM)* that effectively injects the temporal information into subsequent tracking processes.

Like most DETR-family methods [6], we use a ResNet-50 [15] backbone and a Transformer Encoder to produce the image feature of an input frame I^t. As shown in Figure 1, the learnable detect query Q_{det} is fed into the *Detection Decoder* $\mathcal{D}_{det}$ (in Section 3.2) to generate the detect embedding E^t_{det} for the current frame. Afterward, by query-

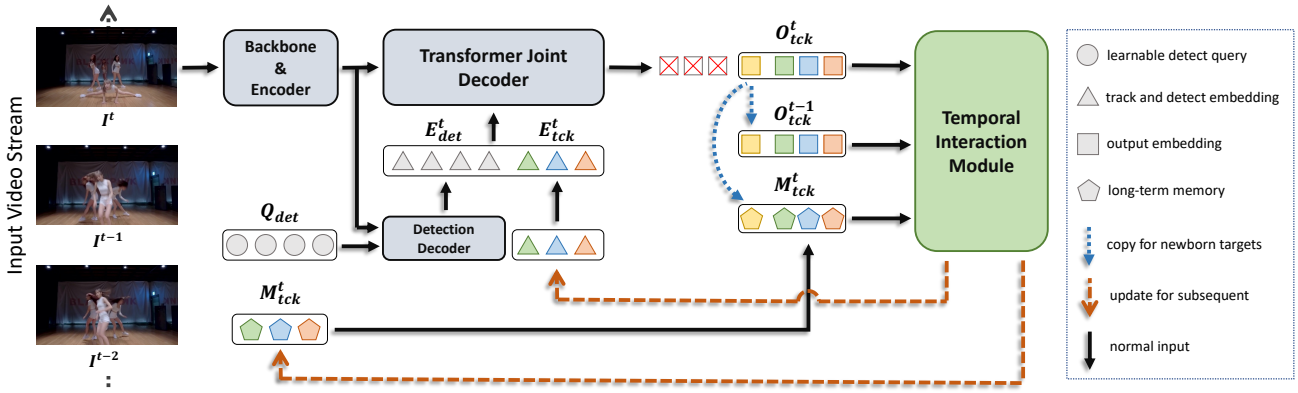


Figure 1. **Overview of MeMOTR.** Like most DETR-based [6] methods, we exploit a ResNet-50 [15] backbone and a Transformer [31] Encoder to learn a 2D representation of an input image. We use different colors to indicate different tracked targets, and the learnable detect query Q_{det} is illustrated in gray. Then the Detection Decoder $\mathcal{D}_{det}$ processes the detect query to generate the detect embedding E^t_{det}, which aligns with the track embedding E^t_{tck} from previous frames. Long-term memory is denoted as M^t_{tck}. The initialization process in the blue dotted arrow will be applied to newborn objects. Our Long-Term Memory and Temporal Interaction Module is discussed in Section 3.3 and 3.4. More details are illustrated in Figure 2.

ing the encoded image feature with $[E^t_{det}, E^t_{tck}]$, the Transformer Joint Decoder $\mathcal{D}_{joint}$ produces the corresponding output $[\hat{O}^t_{det}, \hat{O}^t_{tck}]$. For simplicity, we merge the newborn objects in $\hat{O}^t_{det}$ (yellow box) with tracked objects' output $\hat{O}^t_{tck}$, denoted by O^t_{tck}. Afterward, we predict the classification confidence c^t_i and bounding box b^t_i corresponding to the i^{th} target from the output embeddings. Finally, we feed the output from adjacent frames $[O^t_{tck}, O^{t-1}_{tck}]$ and the long-term memory M^t_{tck} into the Temporal Interaction Module, updating the subsequent track embedding E^{t+1}_{tck} and long-term memory M^{t+1}_{tck}. The details of our components will be elaborated in the following sections.

3.2. Detection Decoder

In the previous Transformer-based methods [22, 42], the learnable detect query and the previous track query are jointly input to Transformer Decoder from scratch. This simple idea extends the end-to-end detection Transformer [6] to multi-object tracking. Nonetheless, we argue that this design may cause misalignment between detect and track queries. As discussed in numerous works [6, 20], the learnable object query in DETR-family plays a role similar to a learnable anchor with little semantic information. On the other hand, track queries have specific semantic knowledge to resolve their category and bounding boxes since they are generated from the output of previous frames.

Therefore, as illustrated in Figure 1, we split the original Transformer Decoder into two parts. The first decoder layer is used for detection, and the remaining five layers are used for joint detection and tracking. These two decoders have the same structure but different inputs. The Detection Decoder $\mathcal{D}_{det}$ takes the original learnable detect query Q_{det} as input and generates the corresponding detect embedding E^t_{det}, carrying enough semantic information to locate and classify the target roughly. After that, we concatenate the detect and track embedding together and feed them into the Joint Decoder $\mathcal{D}_{joint}$.

3.3. Long-Term Memory

Unlike previous methods [17, 42] that only exploit adjacent frames' information, we explicitly introduce a *long-term memory* M^t_{tck} to maintain longer temporal information for tracked targets. When a newborn object is detected, we initialize its long-term memory with the current output.

It should be noted that in a video stream, objects only have minor deformation and movement in consecutive frames. Thus, we suppose the semantic feature of a tracked object changes only slightly in a short time. In the same way, our long-term memory should also update smoothly over time. Inspired by [28], we apply a simple but effective running average with exponentially decaying weights to update long-term memory M^t_{tck}:

$$\widetilde{M}^{t+1}_{tck} = (1-\lambda)M^t_{tck} + \lambda \cdot O^t_{tck}, \tag{1}$$

where $\widetilde{M}^{t+1}_{tck}$ is the new long-term memory for the next frame. The memory update rate λ is experimentally set to 0.01, following the assumption that the memory changes smoothly and consistently in consecutive frames. We also tried some other values in Table 7.

3.4. Temporal Interaction Module

Adaptive Aggregation for Temporal Enhancement. Issues such as blurring or occlusion are often seen in a video stream. An intuitive idea to solve this problem is using

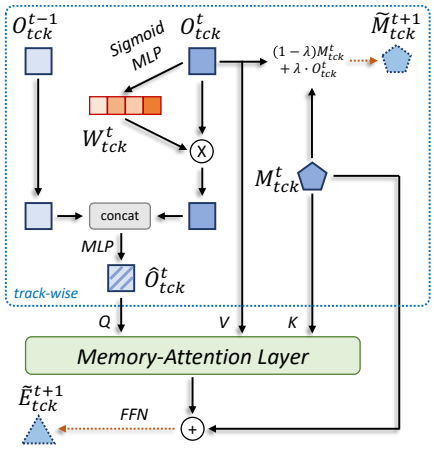


Figure 2. Illustration of **Temporal Interaction Module.** $\widetilde{E}_{tck}^{t+1}$ and $\widetilde{M}_{tck}^{t+1}$ are the prediction of E_{tck}^{t} and M_{tck}^{t} for the next frame, respectively.

multi-frame features to enhance the single-frame representation. Therefore, we fuse the outputs from two adjacent frames with an adaptive aggregation algorithm in our MeMOTR. Due to occlusions and blurring, the output embedding O_{tck}^{t} of the current frame may be unreliable. Thus, as illustrated in Figure 2, we generate a *channel-wise weight* W_{tck}^{t} for each tracked instance to alleviate this problem:

$$W_{tck}^{t} = \text{Sigmoid}(\text{MLP}(O_{tck}^{t})). \tag{2}$$

We multiply this weight W_{tck}^{t} with the current output O_{tck}^{t} and then concatenate the result with O_{tck}^{t-1} from the previous frame. Furthermore, we apply a two-layer MLP to produce the fusion outcome $\widetilde{O}_{tck}^{t}$. This adaptive aggregation enhances the target representation with short-term temporal modeling.

However, we do not use the above channel-wise weight for previous output O_{tck}^{t-1}. As we will discuss in Section 3.5, there is a difference between O_{tck}^{t} and O_{tck}^{t-1}. During inference, we employ a score threshold τ_{tck} to guarantee that O_{tck}^{t-1} is relatively reliable. Therefore, we input it entirely into the subsequent fusion step without the adaptive weight.

Generate Track Embedding. As discussed in Section 3.1, we exploit the track embedding E_{tck}^{t} to produce the location and category of each tracked target. Therefore, generating more reliable and distinguishable track embedding is the key to improving tracking performance. Our processing is illustrated in Figure 2.

Since there are multiple similar objects in the same frame, we believe that learning more discriminative representations is also crucial to the tracker. Thus we employ a Multi-Head Attention [31] structure called *memory-attention layer* to achieve this interaction between different trajectories. Due to the reliability of long-term memory M_{tck}^{t}, we use it as K, and the aggregation $\hat{O}_{tck}^{t}$ and the output embedding O_{tck}^{t} as Q and V, respectively.

After that, we combine long-term memory M_{tck}^{t} and the result of memory-attention layer using addition, then input into an FFN network to predict the subsequent track embedding $\widetilde{E}_{tck}^{t+1}$. As shown in Equation (1), long-term memory is gradually changing over time. Therefore, by incorporating information from long-term memory, the track embedding $\widetilde{E}_{tck}^{t+1}$ avoids abrupt changes that may cause association mistakes in consecutive frames. This design significantly improves the performance of object association, corroborated by ablation experiments shown in Table 6.

3.5. Inference Details

At time step t, we jointly input the learnable detect query Q_{det} and track embedding E_{tck}^{t} ($E_{tck}^{0} = \emptyset$) into our model to produce detection and tracking results, respectively. The detection result with a confidence score of more than τ_{det} will transform into a newborn object.

Target occlusion is a common issue in multi-object tracking task. If a tracked object is lost (confidence $\leq \tau_{tck}$) in the current frame, we do not directly remove its track embedding but mark it as *inactive* trajectory. Afterward, the inactive target will be removed entirely after $\mathcal{T}_{miss}$ frames.

It is worth noting that we do not update the track embedding and long-term memory at every time step for each object. Instead, we choose to update those track embedding with high confidence. The choice of update threshold τ_{next} yields the following formulation for updating:

$$[E_i^{t+1}, M_i^{t+1}] = \begin{cases} [\widetilde{E}_i^{t+1}, \widetilde{M}_i^{t+1}], & c_i^t > \tau_{next} \\ [E_i^t, M_i^t], & c_i^t \leq \tau_{next} \end{cases}, \tag{3}$$

where i is the target index, t is the frame index, and c_i^t is the predicted classification confidence of the i^{th} object at time step t. $\widetilde{E}_i^{t+1}$ and $\widetilde{M}_i^{t+1}$ are the predictions of track embedding and long-term memory, generated by Temporal Interaction Module shown in Figure 2. For simplicity, we set $\tau_{det} = \tau_{tck} = \tau_{next} = 0.5$ in our experiments. $\mathcal{T}_{miss}$ is set to 30, 15 and 10 on DanceTrack, MOT17 and BDD100K, respectively.

4. Experiments

4.1. Datasets and Metrics

Datasets. We mainly evaluate MeMOTR on the DanceTrack [29] dataset since they have more severe association challenges than traditional pedestrian tracking datasets. For a comprehensive evaluation, we also conduct experiments on MOT17 [23] and BDD100K [41].

Metrics. Because of providing a balanced way to measure both detection and association performance explicitly,

we use Higher Order Metric for Evaluating Multi-Object Tracking (HOTA) [21] to evaluate our method, especially analyzing our memory mechanism using Association Accuracy (AssA). We also list the MOTA [2] and IDF1 [25] metrics in our experimental results.

4.2. Implementation Details

Following the settings in MOTR [42], we use several data augmentation methods, such as random resize and random crop. The shorter side of the input image is resized to 800, and the maximum size is restricted to 1536.

We built MeMOTR upon DAB-Deformable-DETR [20] with a ResNet50 [15] backbone and initialize our model with the official DAB-Deformable-DETR [20] weights pre-trained on the COCO [19] dataset. We suggest that the anchor-based position-prior from DAB-Deformable-DETR is quite effective due to the tracked box's smoothness in time and can be further exploited in future works. We also provide the results of our model based on Deformable-DETR [48] for fair comparison in Table 1. Our models are conducted on PyTorch with 8 NVIDIA Tesla V100 GPUs. By using PyTorch gradient checkpoint technology, we implement a memory-optimized version that can also be trained on NVIDIA GPUs with less than 10GB GPU memory. The batch size is set to 1 per GPU, and each batch contains a video clip with multiple frames. Within each clip, video frames are sampled with random intervals from 1 to 10. We use the AdamW optimizer with the initial learning rate of 2.0×10^{-4}. During training, we filter out the tracked target lower than the score threshold $\tau_{update} = 0.5$ and IoU threshold $\tau_{IoU} = 0.5$.

On **DanceTrack** [29], we train MeMOTR for 18 epochs on the train set and drop the learning rate by a factor of 10 at the 12^{th} epoch. Firstly, we use two frames within a clip for training. And then increase the clip frames to 3, 4, and 5 at the 6^{th}, 10^{th}, and 14^{th} epochs, respectively. On **MOT17** [23], due to the small train set (about 5K frames), it is easy to cause overfitting problems. Therefore, we add CrowdHuman [27] validation set to build a joint train set with MOT17 training data. CrowdHuman val set provides about 4K static images. Therefore, we apply random shifts from CenterTrack [46] to generate pseudo trajectories. Finally, we train MeMOTR for 130 epochs, and the learning rate decays by a factor of 10 at the 120^{th} epoch. The initial length of the training video clip is 2 and gradually increases to 3 and 4 at the 60^{th} and 100^{th} epochs, respectively. On **BDD100K** [41], we modify the sampling length at the 6^{th} and 10^{th} epochs and totally train 14 epochs while reducing the learning rate at the 12^{th} epoch.

4.3. Comparison on the DanceTrack Dataset

Since DanceTrack [29] is a dataset with various motions that cannot be modeled by classic linear motion estimation [3, 5], it provides a better choice to verify our tracking performance, especially the association performance.

Methods	HOTA	DetA	AssA	MOTA	IDF1
w/o extra data:					
FairMOT [44]	39.7	66.7	23.8	82.2	40.8
CenterTrack [46]	41.8	78.1	22.6	86.8	35.7
TraDeS [36]	43.3	74.5	25.4	86.2	41.2
TransTrack [30]	45.5	75.9	27.5	88.4	45.2
ByteTrack [43]	47.7	71.0	32.1	89.6	53.9
GTR [47]	48.0	72.5	31.9	84.7	50.3
QDTrack [24]	54.2	80.1	36.8	87.7	50.4
MOTR [42]	54.2	73.5	40.2	79.7	51.5
OC-SORT [5]	55.1	80.3	38.3	**92.0**	54.6
C-BIoU [40]	60.6	**81.3**	45.4	91.6	61.6
MeMOTR* (ours)	63.4	77.0	52.3	85.4	65.5
MeMOTR (ours)	**68.5**	80.5	**58.4**	89.9	**71.2**
with extra data:					
MT_IoT [39]	66.7	84.1	53.0	94.0	70.6
MOTRv2 [45]	69.9	83.0	59.0	91.9	71.7

Table 1. Performance comparison with state-of-the-art methods on the DanceTrack [29] test set. Results for existing methods are from DanceTrack [29]. MeMOTR* means the result based on standard Deformable-DETR.

We compare MeMOTR with the state-of-the-art methods on the DanceTrack [29] test set in Table 1. Our method achieves 68.5 HOTA and gains a vast lead on the AssA metric (58.4 AssA), even surpassing some methods [39] that use additional datasets for training. Due to the limitations of the linear motion estimation module, some tracking-by-detection methods, for example, ByteTrack [43], although they can achieve great detection results (71.0 DetA), still cannot handle complex object association problems (32.1 AssA). However, their MOTA metrics are still high because MOTA overemphasizes detection performance.

Our temporal interaction module, shown in Figure 2, leverages temporal information gracefully and efficiently. Moreover, the separated detection decoder $\mathcal{D}_{det}$ discussed in Section 3.2 alleviates the conflicts between detection and tracking tasks. Therefore, we earn an impressive association performance (58.4 AssA and 71.2 IDF1) and competitive detection performance (80.5 DetA) compared with the state-of-the-art methods. We further prove our components' effectiveness in Section 4.6.

4.4. Comparison on the MOT17 Dataset

In order to make a comprehensive comparison, we also evaluate our method on the classic pedestrian tracking benchmark. Table 2 compares our method with state-of-the-art methods on the MOT17 [23] test set.

Recent tracking-by-detection methods [40, 43] exploit robust detectors (like YOLOX [12]) to achieve really excel-

Methods	HOTA	DetA	AssA	MOTA	IDF1
CNN based:					
Tracktor++ [1]	44.8	44.9	45.1	53.5	52.3
CenterTrack [46]	52.2	53.8	51.0	67.8	64.7
TraDeS [36]	52.7	55.2	50.8	69.1	63.9
QDTrack [24]	53.9	55.6	52.7	68.7	66.3
GTR [47]	59.1	61.6	57.0	75.3	71.5
FairMOT [44]	59.3	60.9	58.0	73.7	72.3
DeepSORT [34]	61.2	63.1	59.7	78.0	74.5
SORT [3]	63.0	64.2	62.2	80.1	78.2
ByteTrack [43]	63.1	64.5	62.0	80.3	77.3
Quo Vadis [10]	63.1	64.6	62.1	80.3	77.7
OC-SORT [5]	63.2	63.2	63.4	78.0	77.5
C-BIoU [40]	64.1	64.8	63.7	81.1	79.7
Transformer based:					
TrackFormer [22]	/	/	/	74.1	68.0
TransTrack [30]	54.1	**61.6**	47.9	74.5	63.9
TransCenter [37]	54.5	60.1	49.7	73.2	62.2
MeMOT [4]	56.9	/	55.2	72.5	69.0
MOTR [42]	57.2	58.9	55.8	71.9	68.4
MeMOTR (ours)	**58.8**	59.6	**58.4**	72.8	**71.5**
Hybird based:					
MOTRv2 [45]	62.0	63.8	60.6	78.6	75.0

Table 2. Performance comparison with state-of-the-art methods on the MOT17 [23] test set. The best performance among the Transformer-based methods is marked in **bold**. MOTRv2 [45] is marked in hybrid since their YOLOX [12] proposals.

Method	mTETA	mHOTA	mLocA	mAssocA	mAssA
QDTrack [24]	47.8	/	45.9	48.5	/
DeepSORT [34]	48.0	/	46.4	46.7	/
MOTR [42]	50.7	37.0	35.8	51.0	47.3
TETer [18]	50.8	/	**47.2**	52.9	/
MeMOTR (ours)	**53.6**	**40.4**	38.1	**56.7**	**52.6**

Table 3. Performance comparison on the BDD100K [41] val set. Results for existing methods are from [18].

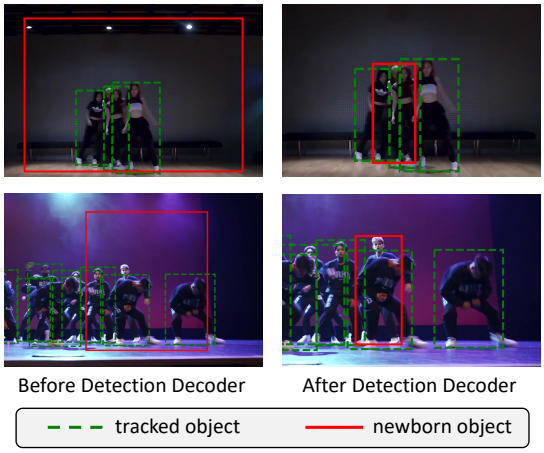


Figure 3. Visualize the anchors of tracked and newborn targets before (left) and after (right) the separated detection decoder $\mathcal{D}_{det}$.

lent detection performance (up to 64.8 DetA). Since performance on MOT17 overemphasizes detection performance, these methods perform immensely well. In this regard, there is still a massive gap in the detection performance of Transformer-based methods [4, 42] because too many dense and small object predictions are involved. In addition, the joint query in a shared Transformer decoder produces tracking and detection simultaneously, which may cause internal conflicts. The detect query is inhibited by the track query in the self-attention [31] structure, limiting the ability to detect newborn objects, especially those close to tracked targets, and vice versa. Because of this, TransTrack [30] achieves significantly better detection performance (61.6 DetA) due to its siamese network structure. This architecture decouples tracking and detection to resolve the above conflict, but its simple post-processing matching algorithm decreases its association performance.

On the other hand, we found that Transformer-based methods [42] suffer from serious overfitting problems in MOT17 [23] because of the tiny train set, which only contains about 5K frames. Although we use an additional CrowdHuman validation set for training mentioned in Section 4.2, severe overfitting still happens. Thus, we get ∼90.0 HOTA and ∼95.0 MOTA on the train set. However, too much additional training data can lead to inductive bias toward static people. Therefore, we argue that the train set of MOT17 is too small to train our model completely.

Eventually, our method slightly improves the performance on the MOT17 test set to 58.8 HOTA. We gain competitive detection accuracy (DetA) compared with other Transformer-based methods. In particular, we improved the performance of object association, which AssA and IDF1 metrics can reflect. As a method that also uses the memory mechanism, our MeMOTR achieves higher AssA and IDF1, surpassing MeMOT [4] by 3.2% and 2.5%, respectively. These experimental results further validate the effectiveness of our method.

4.5. Comparison on the BDD100K Dataset

In order to evaluate our method in multi-category scenarios, we also evaluate our method on the BDD100K [41] val set in Table 3. To better assess multi-class tracking performance, we leverage Tracking-Every-Thing Accuracy (TETA) metric for ranking as they did in BDD100K MOT Challenge 2023. We re-evaluate MOTR [42] on the new metrics using the official model for comparison.

The results show that our method can generalize well to multi-class scenarios and achieve impressive performance

$\mathcal{L}_{\mathcal{D}_{det}}$	HOTA	DetA	AssA	MOTA	IDF1
0	62.1	74.3	52.2	83.1	65.6
1	**63.9**	**74.6**	**55.0**	**83.4**	**67.1**
2	63.2	73.8	54.3	81.9	65.8

Table 4. Ablation experiments on the layers of the separate Detection Decoder, which is denoted as $\mathcal{L}_{\mathcal{D}_{det}}$.

O_{t-1}	W^t_{tck}	HOTA	DetA	AssA	IDF1
		62.3	73.9	52.7	64.6
	✓	62.4	74.5	52.5	64.6
✓		62.7	74.4	53.1	65.3
✓	✓	**63.9**	**74.6**	**55.0**	**67.1**

Table 5. Ablations on different designs of Adaptive Aggregation.

(53.6 TETA), especially in association (56.7 mAssocA), further demonstrating the effectiveness of our proposed method for associating targets.

4.6. Ablation Study

In this section, we study several components of our model, such as the long-term memory, adaptive aggregation, memory-attention layer, and separated detection decoder. Since our main contribution is a better utilization of temporal information, we choose to conduct ablation experiments on DanceTrack [29] due to its more severe object association challenges. On the other hand, DanceTrack [29] has more extensive training data to avoid severe overfitting (about $10\times$ compared with MOT17 [23]). We train our model on the train set and evaluate it on the official val set.

Detection Decoder. For joint tracking and detection, the tracking-by-query paradigm processes detect and track queries in a shared Transformer decoder from scratch. However, the track query has rich semantic content from the previous tracked targets, in contrast to the object query for detection. On the other hand, learnable detect anchors often cover a larger range to find potential targets. In contrast, the anchor of tracked object pays more attention to a small area where the target appeared in the previous frame. This may cause a large gap between the anchors of tracked and newborn objects, as we visualized in Figure 3 (left).

In this paper, we apply a separated Transformer decoder layer to perform preliminary target detection. The output E^t_{det} of this Detection Decoder $\mathcal{D}_{det}$ will be better aligned with the track embedding E^t_{tck} generated by the previous frame to improve the tracking performance. We experimentally confirmed the effectiveness of this design, as shown in Table 4. Using only one separate Detection Decoder layer dramatically improves HOTA and AssA metrics by 1.8% and 2.8%, respectively. However, continuing to increase the layers of the Detection Decoder will reduce the refinement steps of track embeddings, thus slightly weakening the association performance. Furthermore, we visualize the bounding boxes after Detection Decoder in Figure 3 (right). This indicates that $\mathcal{D}_{det}$ is able to locate objects roughly.

M^t_{tck}	*attn*	HOTA	DetA	AssA	IDF1
naïve		61.1	74.2	50.6	63.7
		61.9	73.8	52.1	64.1
✓		62.5	74.2	52.9	64.7
	✓	61.1	74.0	50.7	62.4
✓	✓	**63.9**	**74.6**	**55.0**	**67.1**

Table 6. Ablation study of long-term memory M^t_{tck} and memory-attention layer *attn*. *naïve* means a naive baseline with a single FFN for O^t_{tck} to generate the track embedding $\widetilde{E}^{t+1}_{tck}$.

λ	HOTA↑	DetA↑	AssA↑	IDF1↑	IDsw ↓
0.005	62.6	74.2	53.1	65.1	1383
0.01	**63.9**	74.6	**55.0**	**67.1**	1237
0.02	63.2	75.0	53.5	66.0	**1213**
0.04	63.5	**75.1**	54.0	66.3	1295

Table 7. Ablation study on the long-term memory update ratio λ.

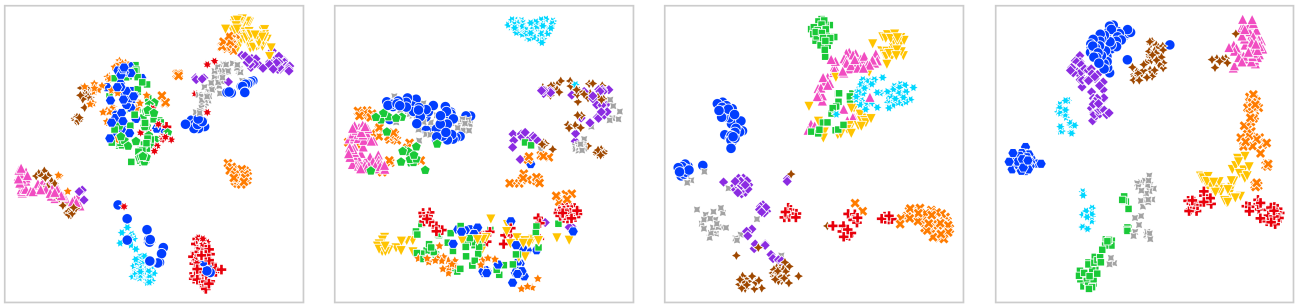
Adaptive Aggregation. In Section 3.4, we design an adaptive aggregation, which dynamically fuses object features from adjacent frames. We ablate this structure in Table 5.

The first two results only use the current output O^t_{tck} to generate the temporal aggregation $\hat{O}^t_{tck}$. In contrast, the next two lines fuse the previous output $\hat{O}^{t-1}_{tck}$ into $\hat{O}^t_{tck}$. Introducing O^{t-1}_{tck} provides additional object features from neighboring frames, thus improving tracking performance. We suppose this offers a complementary feature augmentation that can combat video ambiguity and uncertainty.

Furthermore, we explore the impact of the dynamic weight W^t_{tck}. As shown in Table 5, it only provides a little boost without O^{t-1}_{tck} from the previous. We explain that dynamic W^t_{tck} leads to missing information without complementary features from previous O^{t-1}_{tck}. The result of the last row shows that utilizing both dynamic weight W^t_{tck} and previous output O^{t-1}_{tck} produces significantly better performance, with $+2.6\%$ HOTA and $+2.3\%$ AssA.

Long-Term Memory. We propose a long-term memory in Section 3.3 to utilize longer temporal information and further inject it into subsequent track embedding to augment the object feature.

We explore the impact of long-term memory M^t_{tck} and show the experimental results in Table 6. For a more comprehensive comparison, we also experiment with another track embedding generation structure that removes the



(a) *w/o memory* and *w/o attention* (b) *w/o memory* and *w/ attention* (c) *w/ memory* and *w/o attention* (d) *w/ memory* and *w/ attention* (ours)

Figure 4. **Visualization of Track Embedding** E^t_{tck} (the first 50 frames in sequence dancetrack0063) from different structure designs by using t-Distributed Stochastic Neighbor Embedding (t-SNE). Track embeddings for different tracked targets (IDs) are marked in different colors and shapes. Our design 4(d) helps the model learn a more stable and distinguishable representation for the track embedding. Corresponding tracking performance is shown in Table 6.

memory-attention layer by passing the temporal aggregation $\hat{O}^t_{tck}$ directly to an FFN network.

Our experimental results show that utilizing long-term memory produces a better association performance, with $+0.8\%$ and $+4.3\%$ AssA for *w/o* and *w/ memory-attention layer*. The injection of long-term memory significantly stabilizes and augments the identity information of each track embedding, as visualized in Figure 4(c) and 4(d).

Memory Attention. In Table 6, we also ablate memory-attention layer. It shows that by using memory-attention layer, our MeMOTR achieves much better performance (63.9 *vs.* 62.5 HOTA), especially improving AssA by 2.1%. This attention layer establishes interactions between different trajectories that help the track embedding learns discriminative features. However, without long-term memory M^t_{tck}, memory-attention layer produces worse performance (-1.4% AssA and -1.7% IDF1). We explain that the track embedding without memory augmentation is unstable. Therefore, interacting with such unreliable information can be counterproductive, as visualized in Figure 4(b).

Memory Update Rate. Here, we explored the impact of long-term memory update rate λ on the tracking performance in Equation 1. As shown in Table 7, when progressively increasing the λ_L from 0.005 to 0.04, our model achieves the highest HOTA score at $\lambda = 0.01$ while DetA score is decreasing slightly. We suggest the update rate λ may be a hyperparameter that needs to be chosen according to different datasets. For example, scenarios with plenty of target non-rigid deformation may need a higher memory update rate to adapt to the rapidly changing features.

4.7. Limitations

Although our MeMOTR brings a significant improvement in association performance, detection performance is still a drawback, especially in crowded scenarios (like MOT17 [23]). During experiments, we observed that sometimes newborn objects are suppressed by tracked targets in the self-attention structure, which leads to reduced detection performance. Therefore, resolving this conflict is a crucial challenge for the joint tracking paradigm. This may help improve the detection capabilities, which can boost the overall tracking performance of the model, as studied in [45]. In addition, in pedestrian tracking, existing datasets are still limited in size and diversity. We suggest that training with other simulation datasets (like MOTSynth [11]) may alleviate the overfitting problem of our model and achieve better tracking performance.

5. Conclusion

We have proposed MeMOTR, an end-to-end long-term memory-augmented Transformer for multi-object tracking. Our method builds a stable long-term memory for each tracked object and exploits this memory to augment the representation of track embedding, thus improving its association performance. Furthermore, by leveraging a memory-attention layer, our model makes different targets more distinguishable. As a result, our approach achieves the state-of-the-art performance on MOT benchmarks, especially in scenes with irregular motion patterns. Extensive ablation experiments and visualizations demonstrate the effectiveness of our components. We hope that future work will pay more attention to the use of long-term temporal information for object tracking.

Acknowledgements. This work is supported by National Key R&D Program of China (No. 2022ZD0160900), National Natural Science Foundation of China (No. 62076119, No. 61921006), Fundamental Research Funds for the Central Universities (No. 020214380091, No. 020214380099), and Collaborative Innovation Center of Novel Software Technology and Industrialization. Besides, Ruopeng Gao would like to thank Muyan Yang for her social support.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *ICCV*, pages 941–951. IEEE, 2019. 6

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008, 2008. 5

[3] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016. 1, 2, 5, 6

[4] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *CVPR*, pages 8080–8090. IEEE, 2022. 2, 6

[5] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric SORT: rethinking SORT for robust multi-object tracking. *CoRR*, abs/2203.14360, 2022. 1, 2, 5, 6

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. 2, 3

[7] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 215–230. Springer, 2012. 1

[8] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. In *ICCV*, 2023. 1

[9] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian D. Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *CoRR*, abs/2003.09003, 2020. 2

[10] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? In *NeurIPS*, 2022. 2, 6

[11] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljosa Osep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *ICCV*, pages 10829–10839. IEEE, 2021. 8

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 1, 2, 5, 6

[13] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *CVPR Workshops*, pages 1711–1721. Computer Vision Foundation / IEEE Computer Society, 2018. 1

[14] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 2, 3, 5

[16] Weiming Hu, Tieniu Tan, Liang Wang, and Stephen J. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Syst. Man Cybern. Part C*, 34(3):334–352, 2004. 1

[17] Bruno Korbar and Andrew Zisserman. End-to-end tracking with a multi-query transformer. *CoRR*, abs/2210.14601, 2022. 2, 3

[18] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E. Huang, and Fisher Yu. Tracking every thing in the wild. In *ECCV (22)*, volume 13682 of *Lecture Notes in Computer Science*, pages 498–515. Springer, 2022. 6

[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 5

[20] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: dynamic anchor boxes are better queries for DETR. In *ICLR*. OpenReview.net, 2022. 3, 5

[21] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021. 5

[22] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, pages 8834–8844. IEEE, 2022. 1, 2, 3, 6

[23] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016. 1, 2, 4, 5, 6, 7, 8

[24] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173. Computer Vision Foundation / IEEE, 2021. 5, 6

[25] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops (2)*, volume 9914 of *Lecture Notes in Computer Science*, pages 17–35, 2016. 5

[26] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, pages 6036–6046. Computer Vision Foundation / IEEE Computer Society, 2018. 1

[27] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *CoRR*, abs/1805.00123, 2018. 5

[28] Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 2246–2252. IEEE Computer Society, 1999. 1, 3

[29] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, pages 20961–20970. IEEE, 2022. 1, 2, 4, 5, 7

[30] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *CoRR*, abs/2012.15460, 2020. 2, 5, 6

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3, 4, 6

[32] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV (11)*, volume 12356 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2020. 1, 2

[33] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2

[34] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 2, 6

[35] Sanghyun Woo, Kwanyong Park, Seoung Wug Oh, In So Kweon, and Joon-Young Lee. Tracking by associating clips. In *ECCV (25)*, volume 13685 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2022. 2

[36] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, pages 12352–12361. Computer Vision Foundation / IEEE, 2021. 5, 6

[37] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *CoRR*, abs/2103.15145, 2021. 6

[38] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV (21)*, volume 13681 of *Lecture Notes in Computer Science*, pages 733–751. Springer, 2022. 2

[39] Feng Yan, Zhiheng Li, Weixin Luo, Zequn jie, Fan Liang, Xiaolin Wei, and Lin Ma. Multiple object tracking challenge technical report for team mt_iot, 2022. 5

[40] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *WACV*, pages 4788–4797. IEEE, 2023. 1, 5, 6

[41] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2633–2642. Computer Vision Foundation / IEEE, 2020. 2, 4, 5, 6

[42] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: end-to-end multiple-object tracking with transformer. In *ECCV (27)*, volume 13687 of *Lecture Notes in Computer Science*, pages 659–675. Springer, 2022. 1, 2, 3, 5, 6

[43] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV (22)*, volume 13682 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 2022. 1, 2, 5, 6

[44] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129(11):3069–3087, 2021. 2, 5, 6

[45] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pre-trained object detectors, 2022. 5, 6, 8

[46] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV (4)*, volume 12349 of *Lecture Notes in Computer Science*, pages 474–490. Springer, 2020. 5, 6

[47] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, pages 8761–8770. IEEE, 2022. 2, 5, 6

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*. OpenReview.net, 2021. 5